

Web servers

Simon Domke

Practical course: The bioinformatics lab (SoSe 2012)

Task overview

1. Set up a web server
2. Create test pages
 1. PHP
 2. CGI
3. Install LDAP management tools
4. Secure connection
5. Authentication against LDAP

Apache setup

- default procedure

```
apt-get install apache2
```

- and for executing PHP

```
apt-get install libapache2-mod-php5
```

Test pages (PHP)

- Apache document root: /var/www (default)
- create file /var/www/test.php

```
<form method="get">
<input name="name" type="text" length="10" />
<input type="submit" value="Sag hallo" />
</form>
<?php
print "Hello " . $_GET['name'];
phpinfo();
```

Back

phpinfo() LDAP Account Ma...

Sag hallo

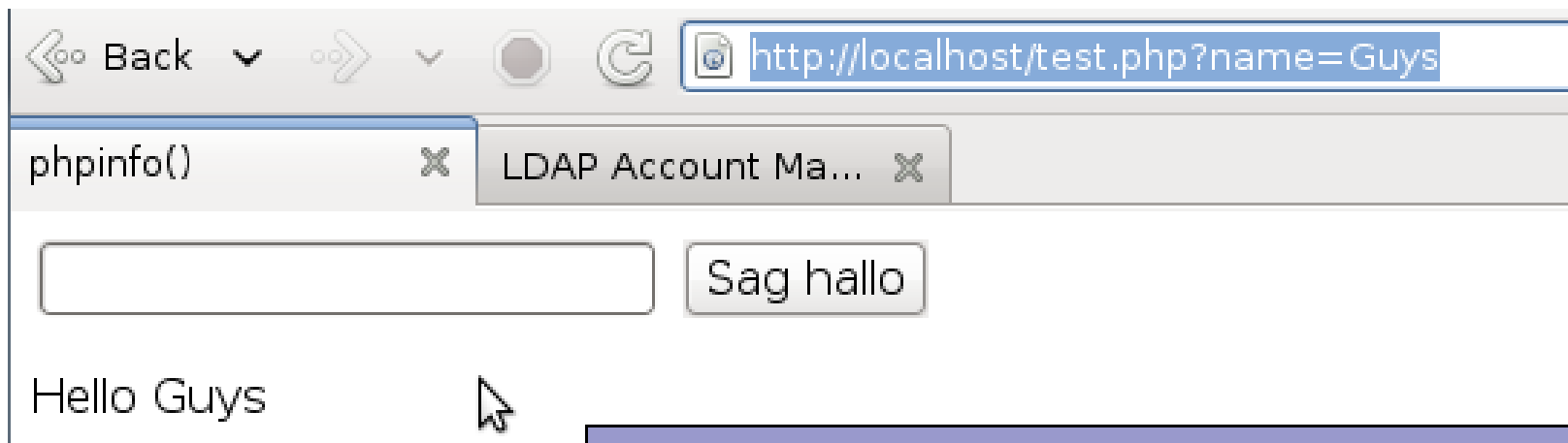
Hello

PHP Version 5.3.3-7+squeeze9

System	Linux simon 2.6.32-5-amd64 #1 SMP Thu Mar 22 17:26:33 UTC 2012 x86_64
Build Date	May 8 2012 10:26:51
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/gd.ini, /etc/php5/apache2/conf.d/ldap.ini, /etc/php5/apache2/conf.d/pdo.ini, /etc/php5/apache2/conf.d/suhosin.ini
PHP API	20090626
PHP Extension	20090626
Zend Extension	220090626

Test pages (PHP)

- enter string „Guys“ into form and click the button



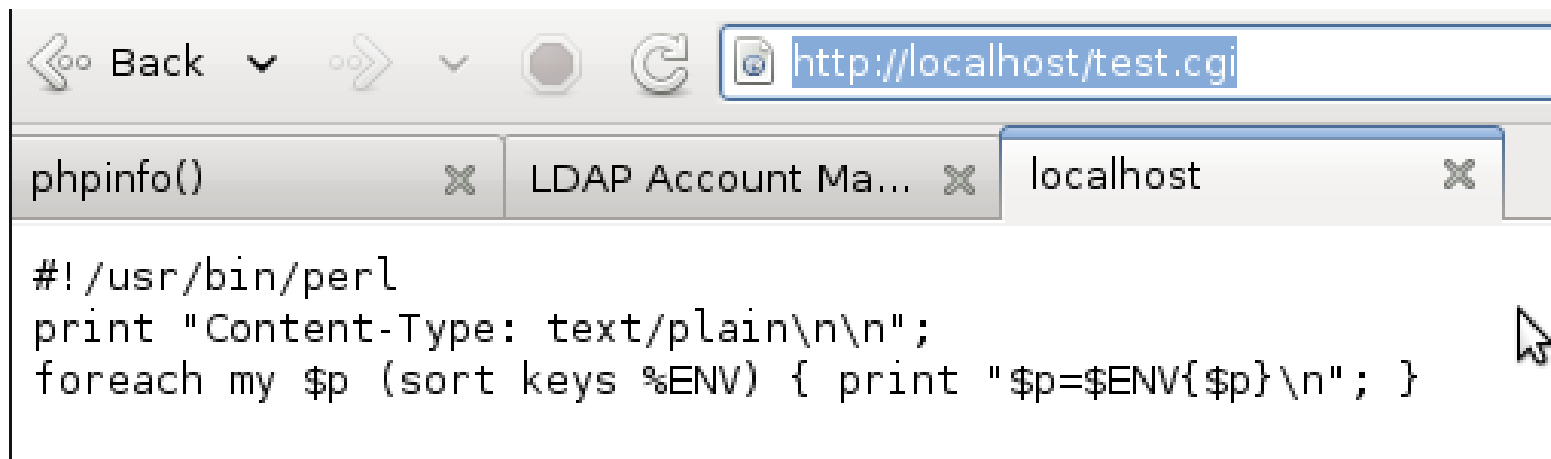
Test pages (CGI)

- create file /var/www/test.cgi

```
#!/usr/bin/perl
print "Content-Type: text/plain\n\n";
foreach my $p (sort keys %ENV) { print "$p=$ENV{$p}\n"; }
```

- chmod 755 /var/www/test.cgi

Test pages (CGI)



A screenshot of a web browser window. The address bar shows `http://localhost/test.cgi`. The browser has three tabs: `phpinfo()`, `LDAP Account Ma...`, and `localhost`. The main content area displays the output of a Perl script, which prints the content type and environment variables. A mouse cursor is visible on the right side of the output text.

```
#!/usr/bin/perl
print "Content-Type: text/plain\n\n";
foreach my $p (sort keys %ENV) { print "$p=$ENV{$p}\n"; }
```

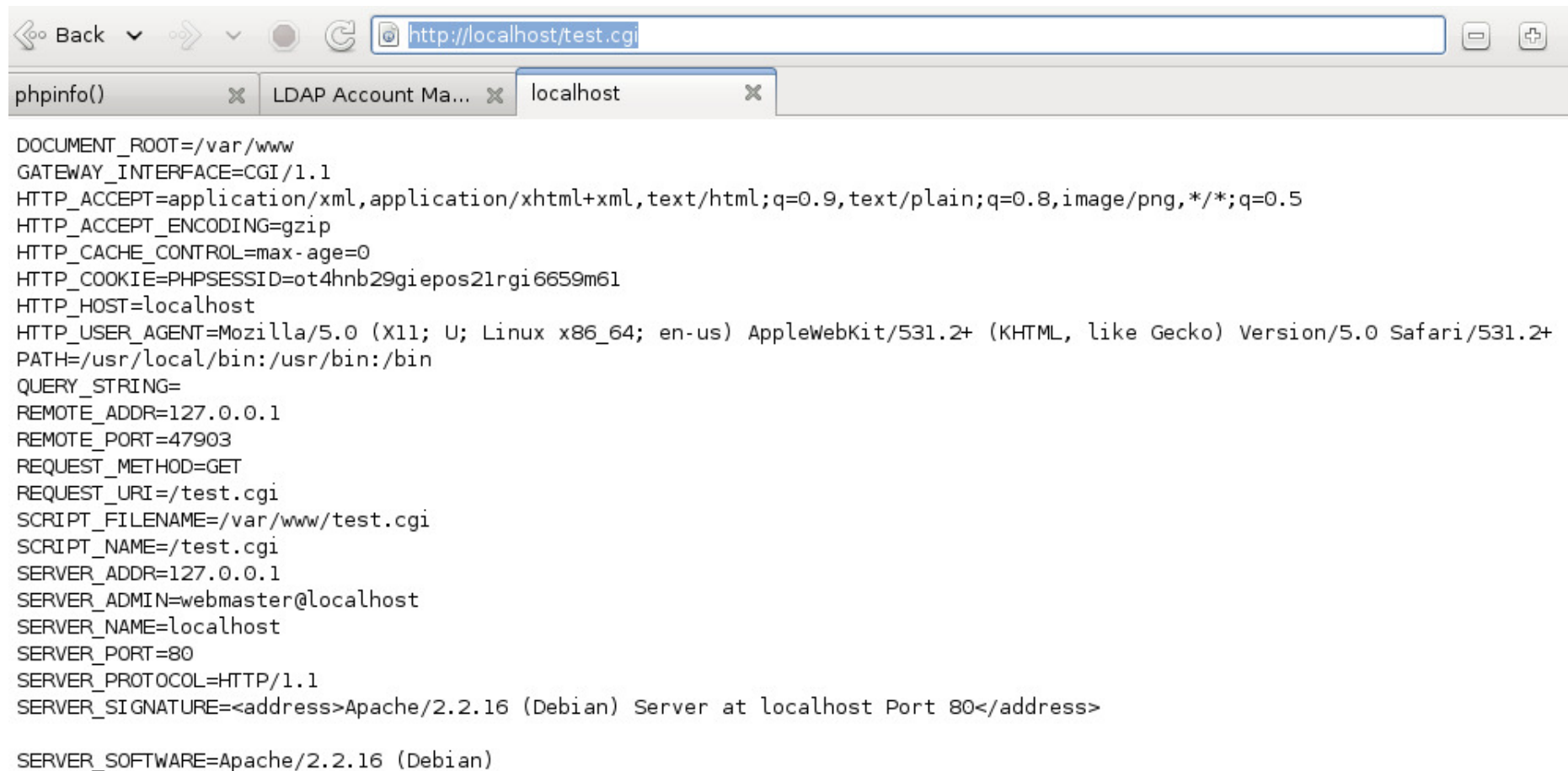
Test pages (CGI)

- obviously Perl code is not executed
- tell Apache to do so (in /etc/apache2/sites-available/default)

```
<Directory /var/www/>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    allow from all
</Directory>
```

```
<Directory /var/www/>
    Options Indexes FollowSymLinks MultiViews +ExecCGI
    AllowOverride None
    Order allow,deny
    allow from all
    AddHandler cgi-script .cgi
</Directory>
```

Test pages (CGI)



```
DOCUMENT_ROOT=/var/www
GATEWAY_INTERFACE=CGI/1.1
HTTP_ACCEPT=application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
HTTP_ACCEPT_ENCODING=gzip
HTTP_CACHE_CONTROL=max-age=0
HTTP_COOKIE=PHPSESSID=ot4hnb29giepos2lrgi6659m61
HTTP_HOST=localhost
HTTP_USER_AGENT=Mozilla/5.0 (X11; U; Linux x86_64; en-us) AppleWebKit/531.2+ (KHTML, like Gecko) Version/5.0 Safari/531.2+
PATH=/usr/local/bin:/usr/bin:/bin
QUERY_STRING=
REMOTE_ADDR=127.0.0.1
REMOTE_PORT=47903
REQUEST_METHOD=GET
REQUEST_URI=/test.cgi
SCRIPT_FILENAME=/var/www/test.cgi
SCRIPT_NAME=/test.cgi
SERVER_ADDR=127.0.0.1
SERVER_ADMIN=webmaster@localhost
SERVER_NAME=localhost
SERVER_PORT=80
SERVER_PROTOCOL=HTTP/1.1
SERVER_SIGNATURE=<address>Apache/2.2.16 (Debian) Server at localhost Port 80</address>

SERVER_SOFTWARE=Apache/2.2.16 (Debian)
```

Task overview

1. Set up a web server
2. Create test pages
 1. PHP
 2. CGI
3. Install LDAP management tools
4. Secure connection
5. Authentication against LDAP

Install LAM ... (and Apache)

- I was lazy and knew the dependencies of the package „ldap-account-manager“
 - apache2
 - libapache2-mod-php5
 - several more (libjs-jquery, php-fpdf, php5, php5-ldap, ...)
- installed only package LAM:

```
apt-get install ldap-account-manager
```

Configuring LAM

- browse „localhost/lam“ -> ***LAM configuration***
- reset master password in ***general settings***
- under ***server profiles*** remove standard profile „lam“ and add my own new one: „simon“
- then edit the profile

Editing LAM profile

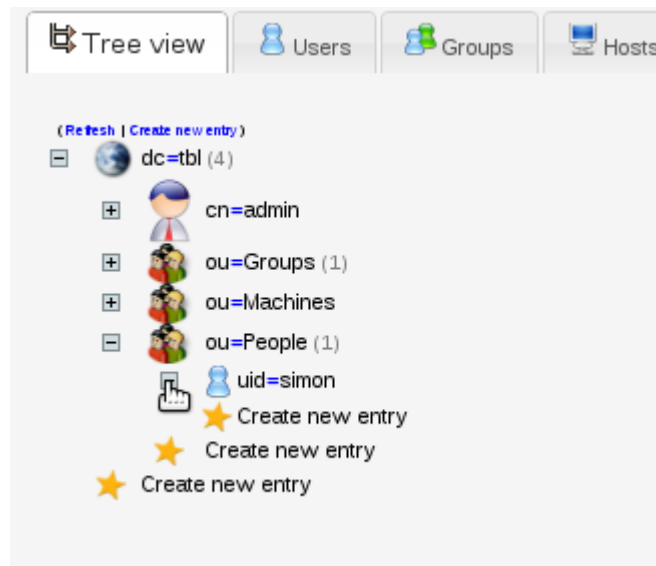
- general settings:
 - address: ldap://localhost:389
 - tree suffix: dc=tbl
 - login is fixed list, edit username: cn=admin,dc=tbl
- account types
 - change LDAP suffix for each entry, e.g. for users:

ou=People,dc=tbl

- click save and login (maybe change the profile first) and let LAM create its entries

LAM usage

- install and config worked, tree is view- and editable



Task overview

1. Set up a web server
2. Create test pages
 1. PHP
 2. CGI
3. Install LDAP management tools
4. Secure connection
5. Authentication against LDAP

Generate certificate (CA)

Create CA (as superuser)

Create a new CA

Name (for local storage):

Data for CA Certificate

Common Name (for the CA):

Country Name (2 letter code):

Password (needed for signing):

Password (confirmation):

State or Province Name:

Locality Name (eg. city):

Organization Name (eg. company):

Organizational Unit Name (eg. section):

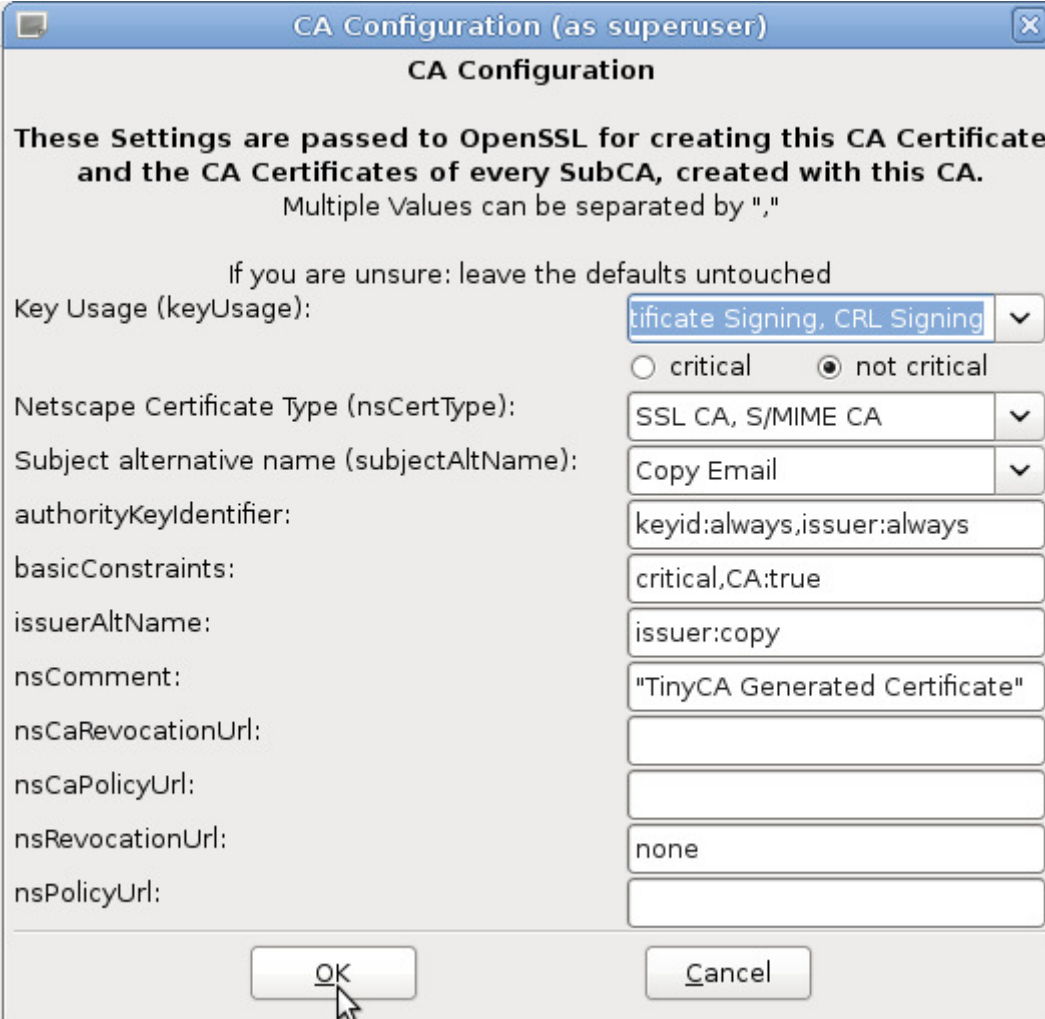
eMail Address:

Valid for (Days):

Keylength: ☐ 1024 ☐ 2048 ☒ 4096

Digest: ☒ SHA-1 ☐ MD2 ☐ MDC2 ☐ MD4 ☐ MD5 ☐ RIPEMD-160

Generate certificate (CA) cont.



CA Configuration (as superuser)

CA Configuration

These Settings are passed to OpenSSL for creating this CA Certificate and the CA Certificates of every SubCA, created with this CA.
Multiple Values can be separated by ","

If you are unsure: leave the defaults untouched

Key Usage (keyUsage):
☐ critical ☒ not critical

Netscape Certificate Type (nsCertType):

Subject alternative name (subjectAltName):

authorityKeyIdentifier:

basicConstraints:

issuerAltName:

nsComment:

nsCaRevocationUrl:

nsCaPolicyUrl:

nsRevocationUrl:

nsPolicyUrl:

Install CA

- `cp /root/.TinyCA/simon/cacert.pem /usr/share/ca-certificates/simon.crt`
- `chmod a+r /usr/share/ca-certificates/simon.crt`
- `dpkg-reconfigure ca-certificates`
- `ln -s /usr/share/ca-certificates/simon.crt /var/www`

```
ca-certificates configuration
This package may install new CA (Certificate Authority) certificates when upgrading. You may want to check
such new CA certificates and select only certificates that you trust.

- yes: new CA certificates will be trusted and installed;
- no : new CA certificates will not be installed by default;
- ask: prompt for each new CA certificate.

Trust new certificates from certificate authorities?

yes
no
ask

<Ok>
```

```
[*] signet.pl/signet_ts1_pem.crt
[*] simon.crt
[*] spi-inc.org/spi-ca-2003.crt
[*] spi-inc.org/spi-cacert-2008.crt
```

Create server certificate

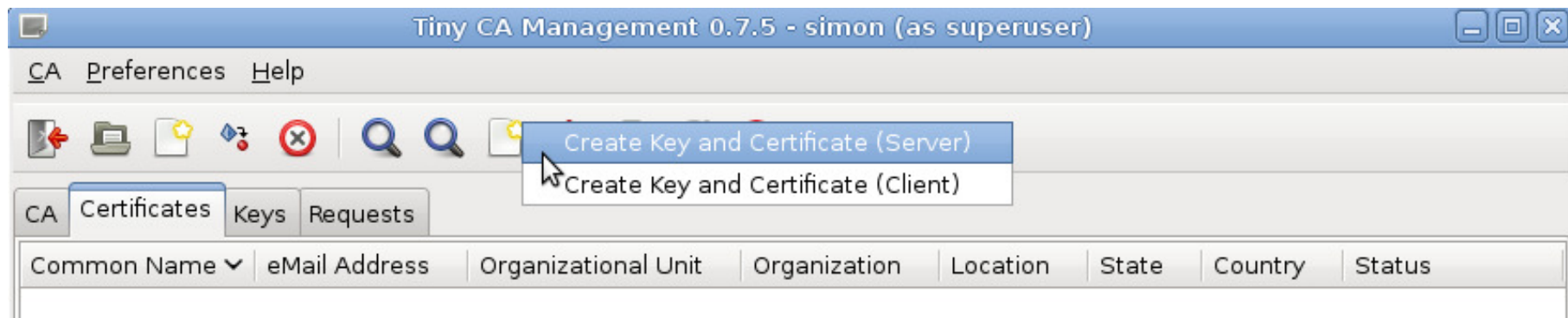
- set TinyCA properties

Subject alternative name (subjectAltName):

Ask User

☐ IP Address ☐ DNS Name ☒ raw

- create certificate



Create server certificate (cont.)

Create Request (as superuser)

Create a new Certificate Request

Common Name (eg, your Name,
your eMail Address
or the Servers Name):

eMail Address:

Password (protect your private Key):

Password (confirmation):

Country Name (2 letter code):

State or Province Name:

Locality Name (eg. city):

Organization Name (eg. company):

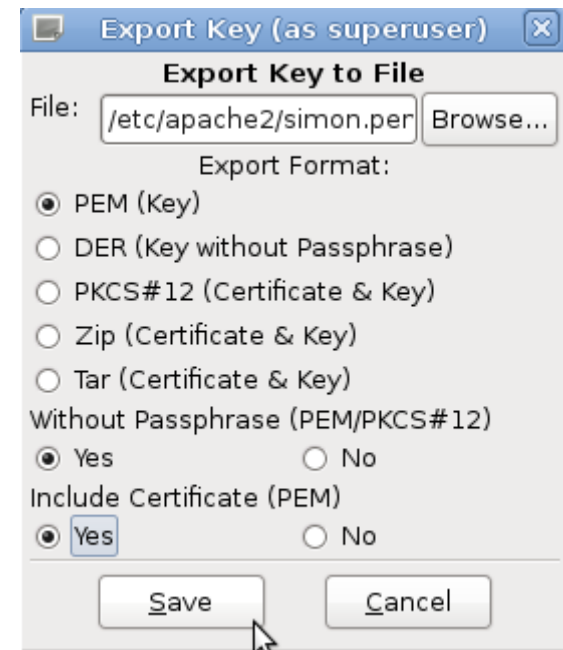
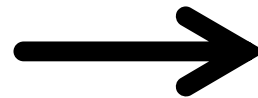
Organizational Unit Name (eg. section):

Keylength: ☒ 4096 ☐ 1024 ☐ 2048

Digest: ☒ SHA-1 ☐ MD2 ☐ MDC2 ☐ MD4 ☐ MD5 ☐ RIPEMD-160

Algorithm: ☒ RSA ☐ DSA

Create server certificate & export key



Edit Apache config

- run:

```
a2enmod ssl rewrite  
a2ensite default-ssl  
service apache2 restart
```

- edit /etc/apache2/sites-available/default

```
<VirtualHost *:80>  
    ServerAdmin webmaster@localhost  
  
    RewriteEngine on  
    RewriteCond %{HTTPS} !=on  
    RewriteRule ^(.*) https://%{SERVER_NAME}/$1 [R,L]
```

Edit Apache config (cont.)

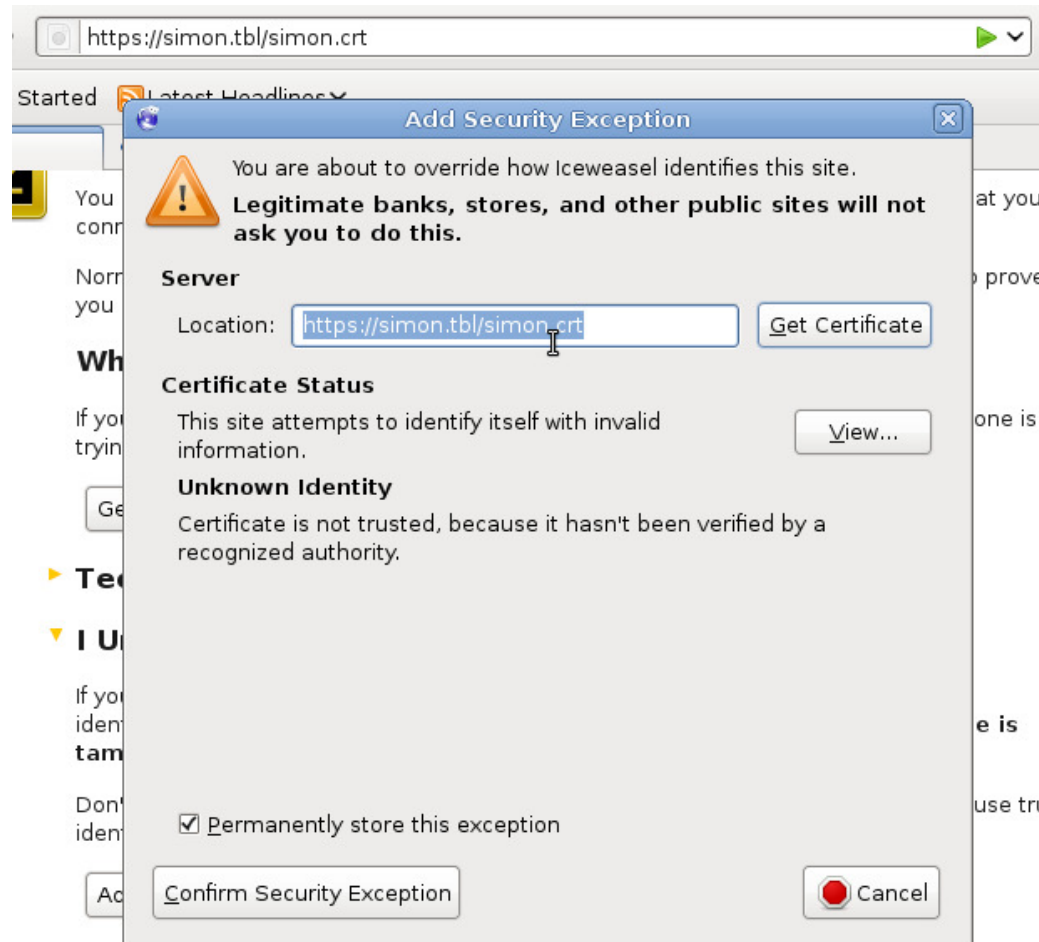
- vi /etc/apache2/sites-available/default-ssl

```
SSLCertificateFile    /etc/apache2/simon.pem  
SSLCertificateKeyFile /etc/apache2/simon.pem
```

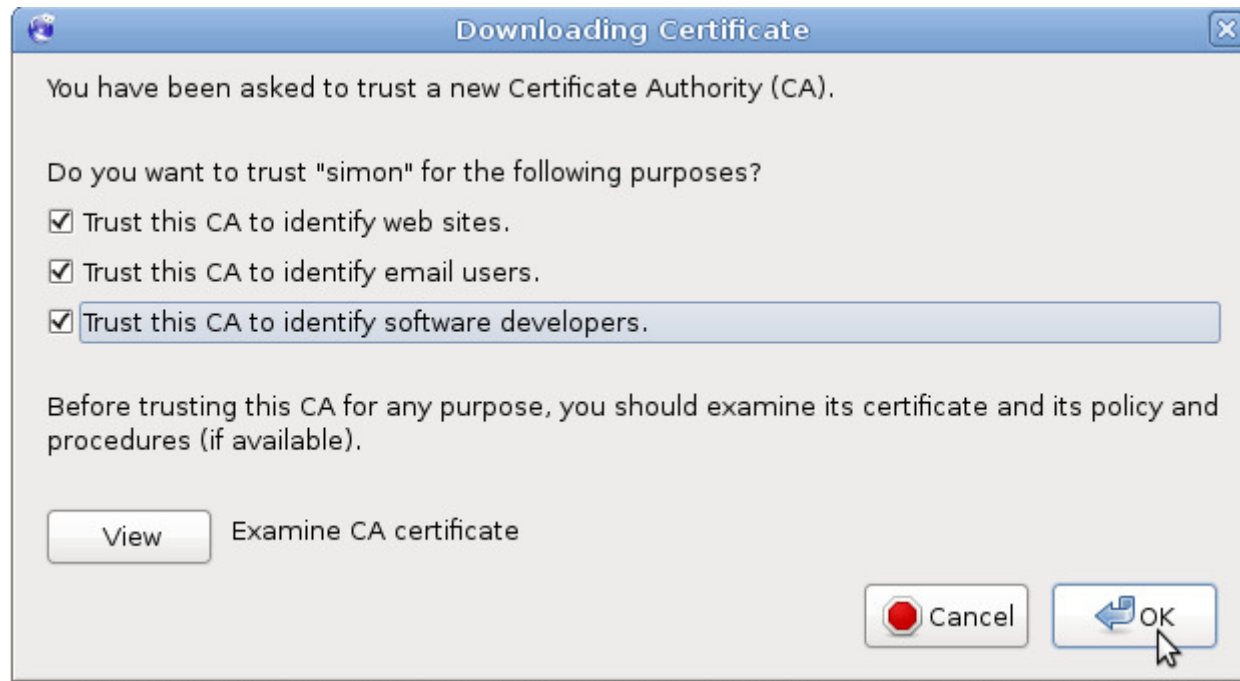
- and copy-paste „<Directory /var/www>“ block from site „default“ into this site

Import certificate in browser

- browse to „http://simon.tbl/simon.crt“



Import certificate in browser



Test with browser



Summary

- all requests on port 80 are now redirected to port 443
- self signed certificate is used for encrypting communication

Task overview

1. Set up a web server
2. Create test pages
 1. PHP
 2. CGI
3. Install LDAP management tools
4. Secure connection
5. Authentication against LDAP

Edit Apache conf

- vi /etc/apache2/sites-available/default-ssl

```
<Directory /var/www/>
    Options Indexes FollowSymLinks MultiViews +ExecCGI
    AllowOverride None
    Order allow,deny
    allow from all
    AddHandler cgi-script .cgi
    AuthType Basic
    AuthName "Please auth"
    AuthBasicProvider ldap
    AuthzLDAPAuthoritative on
    AuthLDAPURL ldap://localhost/dc=tbl?uid?sub
    Require valid-user
</Directory>
```

Test the configuration

- browse again to „simon.tbl/test.php“
- authentication requirement is issued
- enter any valid user in the LDAP database

